FIG. 15 is a block diagram of a system for associative processing in accordance with one embodiment;

FIG. 16 is a block diagram of a system for implementing behavioral operations in accordance with one embodiment of the invention;

FIG. 17 is a block diagram of a system for implementing behavioral operations in accordance with one embodiment of the invention;

FIG. 18 is an example of a behavioral operation;

FIG. 19 is a flow chart of the steps used in a method of behavioral operation of a data document in accordance with one embodiment of the invention; and

FIG. 20 is a flow chart of the steps used in a method of behavioral operation of a data document in accordance with one embodiment of the invention.

Detailed Description of the Drawings

A system for implementing behavioral operations includes a search engine connected to an input data. An associative match memory is connected to the search engine. A behavioral operation unit is connected to the associative match memory.

FIG. 1 is a schematic diagram of a sliding window search routine in accordance with one embodiment of the invention. A data block 20 to be searched is represented as B_0, B_1, B_2 - B_n, where B_0 may represent a byte of data. A first window 22 (W_{1-1}) has a search window size of three bytes. The search window size, in one embodiment, is equal to the size of one of the plurality of data strings for which we are searching. Another window 24 (W_{2-1}) has a search window size of five bytes. An associative database (associative memory) 26 consists of a plurality of address $\{X(W_{n-n})\}$ 28. In one embodiment, the transform of each of the plurality of data strings corresponds to one of the addresses 28 of the associative memory 26. In another embodiment, a transform for at least a first portion of each of the plurality of data strings corresponds to one of the addresses 28 of the associative memory 26. In one embodiment, the transform is a cyclical redundancy code for the plurality of data strings or first portion of the plurality of data strings. In another embodiment, the transform is any linear feedback shift register transformation (polynomial code) of the data string. Generally the polynomial code is selected to have as few collisions as possible.

In one embodiment, a transform (icon) is determined for the first window 22 $\{X(W_{1-1})\}$. Then the address 28 in the associative database

equal to the first window transform is queried. The first entry at the address is a match indicator 30. There are three possible states for the match: no match, match (M) and qualified match (QM). When a match occurs this information is passed to a user (operating system) for further processing. When a no match state is found the window slides by one byte for example. This is shown as window W_{2-1} 32. The subscript one means its the first size window (three byte size) and the subscript two means its the second window. Note the window has slid one byte to cover bytes B_1, B_2, B_3. Prior art techniques, such as hashing, would require determining a completely new transform for the bytes B_1, B_2, B_3. The present invention however uses advanced transform techniques for linear feedback shift registers that are explained in the patent application entitled "Method and Apparatus for Generating a Transform"; serial No. 08/613,037; filed March 8, 1997; assigned to the same assignee as the present application and incorporated herein by reference. These advanced transform techniques are also explained in detail with respect to FIGs. 7-11. Using these advanced techniques a transform (first byte icon) is calculated for a first byte of data (B_o). An icon shift function is performed on the first byte icon to form a shifted first byte icon. Note the shifted first byte icon is $X(B_0\ 0\ 0)$ in this case, where 0 0 represents two bytes of zeros. Note that this discussion also assumes that B_0 is the highest order byte.

The shifted first byte icon $X(B_0\ 0\ 0)$ is exclusive ORed with the first icon $X(B_0\ B_1\ B_2)$ to form a seed icon $X(B_1\ B_2)$. Next a second icon $X(B_1\ B_2\ B_3)$ is formed by transforming a new byte of data (B_3) onto the seed icon $X(B_1\ B_2)$. The process of transforming a new byte of data onto